

## **REMARKS**

[0001] Applicant respectfully requests entry of the following remarks and reconsideration of the subject application. Applicant respectfully requests entry of the amendments herein. The remarks and amendments should be entered under 37 CFR. § 1.116 as they place the application in better form for appeal, or for resolution on the merits.

[0002] Applicant respectfully requests reconsideration and allowance of all of the claims of the application. Claims 1-6, 8-9 and 11-21 are presently pending. Claims amended herein are 1-6, 8-9, 14, 20 and 21. Claims withdrawn or cancelled herein are 7 and 10. No new claims are added herein.

### **Formal Request for an Interview**

[0003] If the Examiner's reply to this communication is anything other than allowance of all pending claims, then I formally request an interview with the Examiner. I encourage the Examiner to call me—the undersigned representative for the Applicant—so that we can discuss this matter so as to resolve any outstanding issues quickly and efficiently over the phone.

[0004] Please contact me or my assistant to schedule a date and time for a telephone interview that is most convenient for both of us. While email works great for us, I welcome your call to either of us as well. Our contact information may be found on the last page of this response.

## **Claim Amendments**

[0005] Without conceding the propriety of the rejections herein and in the interest of expediting prosecution, Applicant amends claims 1-6, 8-9, 14, 20 and 21 herein. Applicant amends claims to clarify claimed features. Such amendments are made to expedite prosecution and to more quickly identify allowable subject matter. Such amendments are merely intended to clarify the claimed features, and should not be construed as further limiting the claimed invention in response to the cited references.

## **Substantive Matters**

### **Claim Rejections under §§ 102 and 103**

[0006] The Examiner rejects claims 1-6, 8-19 and 21 under § 102. For the reasons set forth below, the Examiner has not shown that the cited reference anticipates the rejected claims.

[0007] In addition, the Examiner rejects claims 20 under § 103. For the reasons set forth below, the Examiner has not made a prima facie case showing that the rejected claims are obvious.

[0008] Accordingly, Applicant respectfully requests that the § 102 and § 103 rejections be withdrawn and the case be passed along to issuance.

[0009] The Examiner's rejections are based upon the following references alone and in combination:

- **Nagarajayya:** *Nagarajayya, et al.*, US Patent No. 6,125,402 (issued September 26, 2000); and
- **Schmidt:** *Schmidt, et al.*, US Patent No. 6,546,554 (issued April 8, 2003).

## **Overview of the Application**

[0010] The Application describes a technology for a mechanism for executing a hosted application in either a browser-execution environment or as a standalone application. The mechanism enables hosting code in either of plural hosting environments. (Application, Abstract).

## **Cited References**

[0011] The Examiner cites Nagarajayya as the primary reference in the anticipation and obviousness-based rejections. The Examiner cites Schmidt as the secondary reference in the obviousness-based rejections.

### **Nagarajayya**

[0012] Nagarajayya describes a technology for executing a particular form of a multi-purpose program in a distributed computing environment (Nagarajayya, Abstract).

### **Schmidt**

[0013] Schmidt describes a technology for receiving, installing, and launching applications from a browser (Schmidt, Abstract).

## **Anticipation Rejections**

[0014] Applicant submits that the anticipation rejections are not valid because, for each rejected claim, no single reference discloses each and every element of that rejected claim.<sup>1</sup> Furthermore, the elements disclosed in the single reference are not arranged in the manner recited by each rejected claim.<sup>2</sup>

### **Based upon Nagarajayya**

[0015] The Examiner rejects claims 1-6, 8-19 and 21 under 35 U.S.C. § 102(b) as being anticipated by Nagarajayya. Applicant respectfully traverses this rejection. Based on the reasons given below, Applicant asks the Examiner to withdraw the rejection of these claims.

---

<sup>1</sup> "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987); also see MPEP §2131.

<sup>2</sup> See *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990).

Amended Independent Claim 1

**[0016]** Applicant submits that Nagarajayya does not anticipate this claim because it does not show or disclose at least the following features and elements as recited in this claim (with emphasis added):

- “a first hosting environment”
- “a second hosting environment”
- “a code host mechanism *resident within* the client device”
- “a body of code *received by the client* device to be hosted in either the first or the second hosting environment”
- “wherein the body of code is essentially the same for the first hosting environment and the second hosting environment”
- “wherein *the code host mechanism evaluates the body of code to determine the presence of a particular function call within the body of code* wherein:”
  - “*in the event that the body of code includes the particular function call, the code host mechanism operates to cause the body of code to be hosted in the first hosting environment*”
  - “*in the event that the body of code does not include the particular function call, then the code host mechanism operates to cause the body of code to be hosted in the second hosting environment*”

**[0017]** The Examiner indicates (Action, p. 2-3) the following with regard to this claim:

As to claims 1, 9 and 14, Nagarajayya teaches, the invention substa as claimed including a product, method and client device having computer-readable storage medium having computer-executable components, comprising :

a code host mechanism within the client device (FIG. 2: 240);

a body of code within the client device (FIG. 2:214), wherein:

in the event that the body of code includes a function call (FIG. 2: 216; FIG. 3; col. 8, lines 34-56; wherein the main logic code is the function if called determines where to host the code), the code host mechanism operates to cause the body of code to be hosted in a first hosting environment (FIG. 2; col. 7, lines 36-67; wherein a client 202 provides a first hosting environment);

in the event that the body of code does not include the function call, then the code host mechanism operates to cause the body of code to be hosted in a second hosting environment (FIG. 2; col. 8, lines 17-33; wherein a client 206 provides a second hosting environment for the application program).

**[0018]** Nagarajayya does not disclose or describe that the claimed "code host mechanism evaluates the body of code to determine the *presence of a particular function call within the body of code*". Rather, Nagarajayya uses main logic code to determine whether the program framework will function as an applet or application based on a user initiated action.

[0019] Nagarajayya describes:

[A] program framework that contains a main logic code and an integration of applet and application code. The program framework determines whether a user request is for an applet or an application program. The main logic code then directs whether the program framework will function as an applet or application when executed on the client machine. (see Nagarajayya, column 7, lines 36-67 and column 8, lines 17-56).

[0020] In this Action, the Examiner equates the Applicant's claimed "code host mechanism" with Nagarajayya's Fig. 2, item 240 (see Action pp. 2-3). However, Nagarajayya defines item 240 as *client memory* (Nagarajayya, col. 8, l. 8-16). Client memory does not exhibit the claimed features of the claimed "code host mechanism *resident within the client device*" because client memory just stores instructions and data.

[0021] Additionally, the Examiner equates the claimed "*in the event that the body of code includes the particular function call*" with Nagarajayya's Fig. 2, item 216, which is the **main logic code** that directs what will be downloaded to the client machine (Nagarajayya, col. 7, l. 50-52) or whether code should behave as an applet or an application program (Nagarajayya, col. 8, l. 34-37).

[0022] In contrast to the recitation of this claim, Nagarajayya's main logic code does not *evaluate* the claimed "**body of code** to determine the presence of a particular function call **within the body of code**", and then based on this evaluation "cause the body of code to be hosted in the first hosting environment" or "cause the body of code to be hosted in the second hosting environment".



**[0023]** Instead, the main logic code determines whether the integrated applet/application code will act as an applet or an application on the client platform based on a request. The main logic code does not make this determination based on the claimed “presence of a particular function call” within the integrated applet/application code because Nagarajayya’s main logic code makes this determination based on a request (e.g., user request) (see Fig. 3, item 305).

**[0024]** Furthermore, Nagarajayya does not disclose or describe that the claimed “body of code is essentially the same for the first hosting environment and the second hosting environment” because Nagarajayya’s main logic code determines whether the integrated applet/application code will act as an *applet* or an *application*, which are not “essentially the same”.

**[0025]** Consequently, Nagarajayya does not disclose all of the elements and features of this claim. Accordingly, Applicant asks the Examiner to withdraw the rejection of this claim.

Amended Independent Claim 9

**[0026]** Applicant submits that Nagarajayya does not anticipate this claim because it does not show or disclose at least the following features and elements as recited in this claim (with emphasis added):

- “a default hosting environment in the client device”
- “an alternate hosting environment in the client device”
- “an executable file for hosting within the client device, comprising:”
  - “a first portion including computer-executable instructions operative for performing operations”
  - “a second portion including a function call which, when invoked, is operative to cause the executable file to be executed in the alternative hosting environment”
- “a handler component, executing in either the default hosting environment or the alternate hosting environment, to facilitate the launch of the executable file in either the default hosting environment or the alternate hosting environment, wherein:”
  - “in the event that the executable file invokes the function, the executable file is launched in the alternative hosting environment”
  - “in the event that the executable file does not invoke the function, the executable file is launched in the default hosting environment”

**[0027]** As cited by the Office (Action, p. 2-3) in the rejection to this claim, Nagarajayya describes a program framework that contains a main logic code and an integration of applet and application code. The program framework determines whether a user request is for an applet or an application program. The main logic code then directs whether the program framework will function as an applet or application when executed on the client machine. (see Nagarajayya, column 7, lines 36-67 and column 8, lines 17-56).

**[0028]** Nagarajayya does not disclose or describe the claimed "handler component, executing in **either** the *default hosting environment* or the *alternate hosting environment*, to facilitate the launch of the executable file in **either** the *default hosting environment* or the *alternate hosting environment*". Instead, Nagarajayya describes that main logic code operating in a browser environment launches an applet. Likewise, main logic code operating in an application environment launches an application. For example, Nagarajayya's main logic code cannot execute in a browser environment, then launch its code in an application environment (see Nagarajayya, Fig.2). Therefore, Nagarajayya does not disclose or describe the claimed "handler component, executing in **either** the *default hosting environment* or the *alternate hosting environment*, to facilitate the launch of the executable file in **either** the *default hosting environment* or the *alternate hosting environment*".

**[0029]** Additionally, Nagarajayya does not disclose or describe the claimed "in the event that the executable file invokes the function, the executable file is launched in the alternative hosting environment" because Nagarajayya uses main logic code to determine whether the program framework will launch as an *applet* or

*application* based on a request (Nagarajayya, Fig. 3). Nagarajayya does not disclose or describe that the *applet* or *application* invokes a function to determine their hosting environment.

**[0030]** Furthermore, the Office states (Action, p. 2-3) that Nagarajayya describes a first hosting environment (Nagarajayya, figure 2, client 202) and a second hosting environment (Nagarajayya, figure 2, client 206). However, Nagarajayya does not disclose or describe “a default hosting environment in the client device” and “an alternate hosting environment in the client device” because Nagarajayya discloses that each environment is in a different client device.

**[0031]** Consequently, Nagarajayya does not disclose all of the elements and features of this claim. Accordingly, Applicant asks the Examiner to withdraw the rejection of this claim.

## **Obviousness Rejections**

### **Lack of *Prima Facie* Case of Obviousness (MPEP § 2142)**

[0032] Applicant disagrees with the Examiner's obviousness rejections. Arguments presented herein point to various aspects of the record to demonstrate that all of the criteria set forth for making a prima facie case have not been met.

### **Based upon Nagarajayya and Schmidt**

[0033] The Examiner rejects claim 20 under 35 U.S.C. § 103(a) as being unpatentable over Nagarajayya in view of Schmidt. Applicant has rolled certain features of dependent claim 20 into independent claim 14. Therefore, Applicant will address independent claim 14 and dependent claims 15-20 in light of the rejections to claim 20. Applicant respectfully traverses the rejection of these claims and asks the Examiner to withdraw the rejection of these claims.

Amended Independent Claim 14

**[0034]** Applicant submits that Nagarajayya in view of Schmidt does not render this claim unpatentable because they do not show, teach or suggest at least the following features recited in this claim (emphasis added):

- “providing a function that, if called by the executable file, is operative to cause the executable file to be launched in the alternative hosting environment”
- “*invoking a handler component within the shell environment, wherein the handler component determines if the executable file is configured to call the function, wherein in the event the handler component determines that the executable file is configured to call the function, the handler component facilitates the launch of the executable file, wherein:*”
  - “in the event that the executable file calls the function, the executable file is launched in the alternative hosting environment”
  - “in the event that the executable file does not call the function, the executable file is launched in the default hosting environment”
- “wherein a determination of whether to launch the executable file in either the default hosting environment or the alternative hosting environment is based substantially exclusively on the executable file being configured to call the function”

**[0035]** Neither Nagarajayya nor Schmidt, alone or in any combination teach or suggest the claimed “*invoking a handler component within the shell*

*environment, wherein the handler component determines if the executable file is configured to call the function, wherein in the event the handler component determines that the executable file is configured to call the function, the handler component facilitates the launch of the executable file"* because neither Nagarajayya nor Schmidt teach that the claimed "handler component determines if the executable file is configured to call the function ... wherein a determination of whether to launch the executable file in either the default hosting environment or the alternative hosting environment is based substantially exclusively on the executable file being configured to call the function".

**[0036]** Applicant agrees with the Office (Action p. 5) that Nagarajayya does not explicitly teach the executable file being configured to call the function.

**[0037]** However, the Office has failed to show how Schmidt compensates for this deficiency. The Office asserts that Schmidt discloses configuration of an application downloaded on to a client from a browser for selecting the appropriate runtime hosting environment, citing column 4, lines 39-51 and column 6, lines 1-32 which state in part:

In the context of the present invention, a helper application is a stand-alone application that is invoked by a browser when a file of a specified file type is encountered by the browser. Typically, the helper application must be "registered" with the browser for a specified file type, so that the browser knows to invoke the correct helper application when the specified file type is encountered. ... Unlike "plug-ins," helper applications typically execute as separate processes with respect to the browser.

... a new type of helper application, which is called a Java Net Launcher ("JNet") 420 in one embodiment of the resent invention. As will be discussed in more detail below, in one embodiment, the JNet helper

application 420 is registered with browser 410 such that the JNet helper application 420 is invoked when browser 410 encounters a metafile having the new Java Net Launcher (".jnl," or ".JNL") file format. In the context of the present invention, the term "metafile" refers to a file which describes or defines aspects or components of an application.

If multiple JRE's are installed on the client computer 10, the JNet helper application 420 can select the appropriate JRE 430 to use for a given application based on information provided in the JNL metafile. If the appropriate JRE has not been installed on the client computer 10, the JNet helper application 420 can automatically download and install the appropriate JRE from a trusted source.

**[0038]** As shown above, Schmidt teaches that a helper application is a stand-alone application that is invoked by a browser when a file of a specified file type is encountered by the browser. However, the combination of Nagarajayya and Schmidt does not teach that a helper application "determines if the executable file is configured to call the function" as recited in this claim. Additionally, Schmidt does not teach that the claimed "executable file calls the function", because a file downloaded by Schmidt does not call a function to determine which helper application will host the downloaded file.

**[0039]** Schmidt also teaches that a JNet helper application can select the appropriate JRE to use for a given application based on information provided in the JNL metafile. However, the combination of Nagarajayya and Schmidt does not teach that any determination is made based on whether the given application "is configured to call the function". Additionally, in Schmidt, the downloaded file does not call any function to determine any JNet helper application or JRE.



**[0040]** As shown above, the combination of Nagarajayya and Schmidt does not teach or render obvious all of the elements and features of this claim. Accordingly, Applicant asks the Examiner to withdraw the rejection of this claim.

### **Dependent Claims**

**[0041]** In addition to its own merits, each dependent claim is allowable for the same reasons that its base claim is allowable. Applicant requests that the Examiner withdraw the rejection of each dependent claim where its base claim is allowable.

## **Conclusion**

[0042] All pending claims are in condition for allowance. Applicant respectfully requests reconsideration and prompt issuance of the application. If any issues remain that prevent issuance of this application, the **Examiner is urged to contact me before issuing a subsequent Action.** Please call or email me or my assistant at your convenience.

Respectfully Submitted,

Lee & Hayes, PLLC  
Representatives for Applicant

/E. John Fain Reg. No. 60960/ Dated: 8/25/2008

E. John Fain (johnf@leehayes.com; x256)

Registration No. 60,960

John C. Meline (johnm@leehayes.com; x257)

Registration No. 58,2080

Customer No. **22801**

Telephone: (509) 324-9256

Facsimile: (509) 323-8979

www.leehayes.com